

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07234879 A**(43) Date of publication of application: **05 . 09 . 95**

(51) Int. Cl.

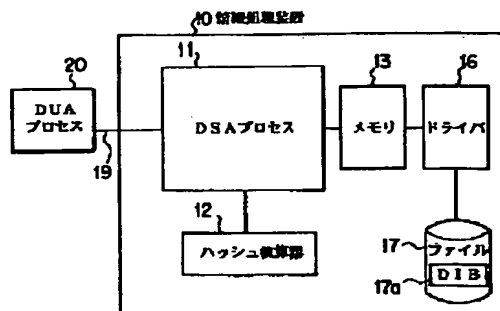
G06F 17/30(21) Application number: **06026646**(22) Date of filing: **24 . 02 . 94**(71) Applicant: **TOSHIBA CORP**(72) Inventor: **HASHIMOTO KOICHI**(54) **INFORMATION PROCESSOR AND DATA BASE
RETRIEVING METHOD**

COPYRIGHT: (C)1995,JPO

(57) Abstract:

PURPOSE: To accelerate the speed of retrieval to a directory information base (DIB).

CONSTITUTION: This device is provided with a DIB 17a, for which logical data structure is tree structure, provided with a peculiar relative identification name (RDN) composed of the combination of attribute types and attribute values so as to store entries for which hash values are added to the attribute value, and a hash computing element 12 for calculating the hash value from the attribute value of the RDN corresponding to the identification name (DN) of the object entry composed of the arrangement of the RDN from the DIB 17a. Further, the device is provided with a driver 16 for reading the entry to be a candidate based on the DN of the object entry and a DSA process 11 for comparing the hash value added to the read entry with the hash value calculated by the hash computing element 12, comparing the attribute values of only the matched hash values, further, comparing the attribute types of the RDN only for the matched entries and retrieving the object entry from the DIB 17a based on the compared result.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平7-234879

(43)公開日 平成7年(1995)9月5日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 17/30		9194-5L	G 0 6 F 15/ 413	3 1 0 A
		9194-5L	15/ 411	3 1 0

審査請求 未請求 請求項の数5 O L (全 8 頁)

(21)出願番号 特願平6-26646

(22)出願日 平成6年(1994)2月24日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 橋本 浩一

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

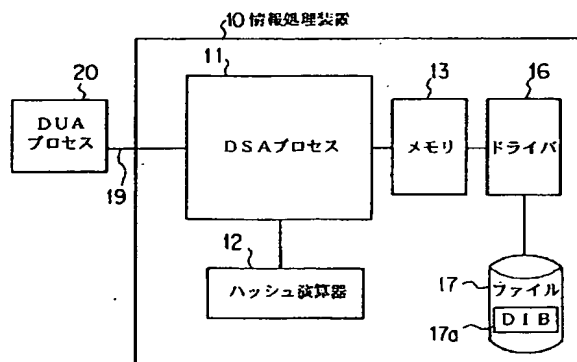
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 情報処理装置及びデータベース検索方法

(57)【要約】

【目的】ディレクトリ情報ベースに対する検索速度を向上させる。

【構成】論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名(RDN)を持ち、属性値にハッシュ値が付加されたエントリが格納されたディレクトリ情報ベース(DIB)17aと、DIB17aからの相対識別名の並びからなる目的エントリの識別名(DN)に対し、相対識別名の属性値からハッシュ値を算出するハッシュ演算器12と、目的エントリの識別名をもとに候補となるエントリを読み出すドライバ16と、読み出されたエントリに付加されたハッシュ値とハッシュ演算器12によって得られたハッシュ値とを比較して一致したものだけ属性値を比較して、さらに一致したエントリについてだけ相対識別名の属性型を比較し、その比較結果に基づいてDIB17aから目的エントリを検索するDSAプロセス11とを具備して構成する。



【特許請求の範囲】

【請求項1】 論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名（R DN）をもつエントリが格納されたディレクトリ情報ベース（D I B）と、前記ディレクトリ情報ベースのルートからの前記相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）をもとに、前記ディレクトリ情報ベースから候補となるエントリを読み出すドライバ手段と、前記ドライバ手段によって読み出された候補となるエントリについて、同エントリの相対識別名の属性値と前記目的エントリの識別名中の対応する相対識別名の属性値とを比較して、一致したエントリについてだけ相対識別名の属性型を比較し、その比較結果に基づいて前記ディレクトリ情報ベースから前記目的エントリを検索する検索手段と、を具備したことを特徴とするディレクトリシステムを実現する情報処理装置。

【請求項2】 論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名（R DN）を持ち、前記属性値にハッシュ値が付加されたエントリが格納されたディレクトリ情報ベース（D I B）と、前記ディレクトリ情報ベースのルートからの相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）に対し、相対識別名の属性値からハッシュ値を算出するハッシュ演算手段と、前記目的エントリの識別名をもとに前記ディレクトリ情報ベースから候補となるエントリを読み出すドライバ手段と、前記ドライバ手段によって読み出された候補となるエントリについて、同エントリに付加されたハッシュ値と前記ハッシュ演算手段によって得られたハッシュ値とを比較して一致したものだけ同エントリの相対識別名の属性値と前記目的エントリの識別名中の対応する相対識別名の属性値とを比較して、さらに一致したエントリについてだけ相対識別名の属性型を比較し、その比較結果に基づいて前記ディレクトリ情報ベースから前記目的エントリを検索する検索手段と、を具備したことを特徴とするディレクトリシステムを実現する情報処理装置。

【請求項3】 前記ドライバ手段によって読み出されたエントリの相対識別名の属性値に付加されたハッシュ値と、対応するエントリを示すアドレスを格納した下位エントリ一覧テーブルを作成し、前記検索手段による比較を行なう際に前記下位ハッシュエントリ一覧テーブルに従ってエントリを探し出すことを特徴とする請求項2記載の情報処理装置。

【請求項4】 論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名（R

DN）をもつエントリが格納されたディレクトリ情報ベース（D I B）から、前記ディレクトリ情報ベースのルートからの前記相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）をもとに目的エントリを検索するデータベース検索方法であって、前記目的エントリの識別名に従って前記ディレクトリ情報ベースから下位エントリの候補を読み出し、読み出されたエントリの相対識別名の属性値と前記目的エントリの識別名中の対応する相対識別名の属性値とを比較し、この比較の結果、属性値が一致したエントリについてだけ相対識別名の属性型を比較し、この比較結果に基づいて前記ディレクトリ情報ベースから前記目的エントリを検索することを特徴とするデータベース検索方法。

【請求項5】 論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名（R DN）を持ち、前記属性値にハッシュ値が付加されたエントリが格納されたディレクトリ情報ベース（D I B）から、前記ディレクトリ情報ベースのルートからの前記相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）をもとに目的エントリを検索するデータベース検索方法であって、前記ディレクトリ情報ベースのルートからの相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）に対し、相対識別名の属性値からハッシュ値を算出し、前記目的エントリの識別名に従って前記ディレクトリ情報ベースから下位エントリの候補を読み出し、読み出された候補となるエントリについて、同エントリに付加されたハッシュ値と前記識別名に対して算出されたハッシュ値とを比較し、この比較の結果、ハッシュ値が一致したエントリについてだけ読み出されたエントリの相対識別名の属性値と前記目的エントリの識別名中の対応する相対識別名の属性値とを比較し、この比較の結果、属性値が一致したエントリについてだけ相対識別名の属性型を比較し、この比較結果に基づいて前記ディレクトリ情報ベースから前記目的エントリを検索することを特徴とするデータベース検索方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、例えばCCITT勧告X. 500で規定されているディレクトリシステムを実現する情報処理装置およびデータベース検索方法に関する。

【0002】

【従来の技術】 例えばCCITT勧告X. 500で規定されているディレクトリシステムは、人やマシンなどの

オブジェクトに関する情報（住所、氏名、電話番号、ネットワークアドレスなど）をエントリとして格納するデータベースをもっており、オブジェクトの情報を検索する機能を利用者に提供するシステムである。

【0003】このシステムにおいて用いられるデータベースは、ディレクトリ情報ベース（以下、DIBと略称する）と呼ばれ、その論理的データ構造が木構造となっており、木の節がエントリとなっている。オブジェクトに関する住所、氏名、電話番号、ネットワークアドレスなどの情報は、属性としてエントリに格納されている。

【0004】DIBのエントリには、上位、下位の関係がある。例えば、図9においては、会社Xの上位エントリは日本である。各エントリは相対識別名（RDN）をもっている。

【0005】相対識別名（RDN）は〔属性型、属性値〕の組み合わせからなっており、例えば会社Xのエントリは〔“組織名”、“会社X”〕という相対識別名（RDN）を持っている。また、ルートから任意のエントリまでの相対識別名（RDN）の並びを識別名（DN）という。

【0006】例えば図9中に示す「橋本浩一」のエントリは、〔〔“国名”、“日本”〕、〔“組織名”、“会社X”〕、〔“一般名”、“橋本浩一”〕〕という識別名（DN）を持っている。利用者が「橋本浩一」に関する情報を検索するときは、この識別名（DN）を指定する。

【0007】ディレクトリシステムは、指定された前述のような識別名（DN）をもとに、ディレクトリ情報ベース（DIB）のルートから「日本」「会社X」とたどり、利用者が検索しようとしている「橋本浩一」のエントリを獲得する。

【0008】

【発明が解決しようとする課題】「日本」の下位エントリには、「会社A」「会社X」「会社B」があり、それぞれ〔“組織名”、“会社A”〕〔“組織名”、“会社X”〕〔“組織名”、“会社B”〕という相対識別名（RDN）を持っている。

【0009】ディレクトリシステムが「日本」のエントリまでたどっており、かつ利用者が指定した識別名（DN）が〔“組織名”、“会社X”〕という相対識別名（RDN）を持っている場合は、その相対識別名（RDN）と上記3つの相対識別名（RDN）を比較し、属性型と属性値の両方が一致した相対識別名（RDN）をもつエントリを探し出す。

【0010】通常、上位エントリが同じエントリ同士は、同じ相対識別名（RDN）の属性型を持つ（上記の場合は“組織名”）。従って、相対識別名（RDN）の比較を行なうとき、属性型の比較で不一致なるケースは非常に少なく、むしろ属性値で不一致となるケースが多いために、結果的に必要以上の比較回数を要し、検索速

度向上の妨げとなっていた。

【0011】また、属性値は文字列であるため、2バイト以上のデータとなるケースが多く、その長さはエントリによってまちまちである。従って、長い文字列からなる属性値をそのまま比較すると、比較に要する時間はその長さに比例して長くなってしまう。

【0012】本発明は前記のような事情を考慮してなされたもので、検索速度を向上させたディレクトリシステムを実現する情報処理装置およびデータベース検索方法を提供することを目的とする。

【0013】

【課題を解決するための手段】本発明は、論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名（RDN）をもつエントリが格納されたディレクトリ情報ベース（DIB）と、前記ディレクトリ情報ベースのルートからの前記相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）をもとに、前記ディレクトリ情報ベースから候補となるエントリを読み出すドライバ手段と、前記ドライバ手段によって読み出された候補となるエントリについて、同エントリの相対識別名の属性値と前記目的エントリの識別名中の対応する相対識別名の属性値とを比較して、一致したエントリについてだけ相対識別名の属性型を比較し、その比較結果に基づいて前記ディレクトリ情報ベースから前記目的エントリを検索する検索手段とを具備したことを特徴とする。

【0014】また本発明は、論理的データ構造が木構造であり、属性型と属性値の組み合わせからなる固有の相対識別名（RDN）を持ち、前記属性値にハッシュ値が付加されたエントリが格納されたディレクトリ情報ベース（DIB）と、前記ディレクトリ情報ベースのルートからの相対識別名の並びからなる、検索対象とする目的エントリの識別名（DN）に対し、相対識別名の属性値からハッシュ値を算出するハッシュ演算手段と、前記目的エントリの識別名をもとに前記ディレクトリ情報ベースから候補となるエントリを読み出すドライバ手段と、前記ドライバ手段によって読み出された候補となるエントリについて、同エントリに付加されたハッシュ値と前記ハッシュ演算手段によって得られたハッシュ値とを比較して一致したものだけ同エントリの相対識別名の属性値と前記目的エントリの識別名中の対応する相対識別名の属性値とを比較して、さらに一致したエントリについてだけ相対識別名の属性型を比較し、その比較結果に基づいて前記ディレクトリ情報ベースから前記目的エントリを検索する検索手段とを具備したことを特徴とする。

【0015】また、前記ドライバ手段によって読み出されたエントリの相対識別名の属性値に付加されたハッシュ値と、対応するエントリを示すアドレスを格納した下位エントリ一覧テーブルを作成し、前記検索手段による比較を行なう際に前記下位ハッシュエントリ一覧テ

ルに従ってエントリを探し出すことを特徴とする。

【0016】

【作用】このような構成によれば、ディレクトリシステムを実現する情報処理装置がエントリを検索するために相対識別名(RDN)の比較を行なうときに、相対識別名(RDN)の比較において属性値で不一致となるケースが多いことに注目し、属性値の比較で一致したものだけに属性型を比較することで、属性型から比較する手段より、全体の比較回数を減らすことができ、検索速度を向上させることができる。

【0017】また、属性値に固定長のハッシュ値をもたせ、その値による検索を行ってから、一致するものの中で属性値の比較を行なうことで、比較対象の属性値の絞り込みを行ない、ハッシュ値を持たせない場合よりも全体としての検索速度を向上させることができる。

【0018】また、下位エントリの候補である相対識別名(RDN)を取り出すために下位エントリ一覧テーブルを使用することによって、高速に下位エントリにアクセスすることができ、全体としての検索速度を向上させることができる。

【0019】

【実施例】以下、図面を参照して本発明の一実施例を説明する。図1は本実施例に係わる情報処理装置の概略構成を示すブロック図である。図1に示すように、本実施例の情報処理装置10は、DSA(ディレクトリサービスエージェント)プロセス11、ハッシュ演算器12、メモリ13、ドライバ16、及びファイル17から構成されている。情報処理装置10(DSAプロセス11)は、通信回線19で接続されているDUAプロセス20から、ディレクトリ情報ベース(以下、DIBと略称する)の検索を行なうための識別名を受信する。

【0020】DSAプロセス11は、メモリ13に直接読み書きが可能であり、通信回線19で接続されているDUAプロセス20を介して送信された、利用者が検索したい目的エントリの識別名(DN)をメモリ13に書き込み、ドライバ16に目的エントリを検索するための下位エントリの候補を読み出させるものである。DSAプロセス11は、識別名(DN)の相対識別名(RDN)に従って比較処理を順次行ない、エントリをたどって目的プロセスを検索していく。

【0021】ハッシュ演算器12は、DSAプロセス11によって受信された、利用者が検索したいエントリの識別名(DN)の、相対識別名(RDN)の属性値からハッシュ値を算出するものである。

【0022】メモリ13は、利用者が検索したい目的エントリの識別名(DN)や、ファイル17に格納されたDIBからドライバ16によって読み出されたエントリ等を格納するためのものである。

【0023】ドライバ16は、DSAプロセス11による制御により、ファイル17に格納されたDIB17a

から、メモリ13に格納された識別名(DN)をもとにエントリの読出し等を実行するものである。

【0024】ファイル17は、情報処理装置10が管理しているDIB17aを格納しており、その一部のエントリがドライバ16によって読み出されてメモリ13に格納される。

【0025】次に、本実施例の動作について説明する。はじめに、ハッシュ演算器12を用いない場合の動作について説明する。まず、利用者が検索したい目的エントリの識別名(DN)が、DUAプロセス20、通信回線19を介してDSAプロセス11に送信される。DSAプロセス11は、受信した識別名(DN)をメモリ13に書き込む。

【0026】図2は、DSAプロセス11によってメモリ13に書き込まれている、目的エントリに対応する識別名(DN)の一例を示している。識別名(DN)は、相対識別名(RDN)の並びであり、{“国名”、“日本”}、{“組織名”、“会社X”}、{“一般名”、“橋本浩一”}が格納されているものとする。なお、ファイル17に格納されたDIB17aは図9に示す構成であるものとする。

【0027】次に、図3に示すフローチャートを参照しながら、DIB17aから目的エントリを取り出す処理について説明する。この際、相対識別名(RDN)の比較処理、すなわち目的エントリの識別名にある相対識別名(RDN)とファイルに格納されている各エントリの相対識別名(RDN)との比較によって、目的エントリを取り出す。

【0028】ここでは、情報処理装置10が目的エントリの識別名(DN)をDUAプロセス20から受信してメモリ13に格納しており、その識別名(DN)をもとにDIB17aに格納されたエントリを、ルートから「日本」を示すエントリまでたどっている状態を考える。

【0029】さて、DSAプロセス11は、ドライバ16を使って、ファイル17のDIB17aから「日本」を示すエントリの、全ての下位エントリの候補をファイル17からメモリ13に読み出す(ステップA1)。

【0030】図4は、メモリ13に格納された下位エントリ21、22、23の候補を示し、「日本」を示すエントリの下位エントリがドライバ16によってファイル17から読み出されている状態を示している。

【0031】図4に示すように、下位エントリ21、22、23は、相対識別名(RDN)を持っており、それぞれ{“組織名”、“会社A”}、{“組織名”、“会社X”}、{“組織名”、“会社B”}である。また、下位エントリ21、22、23は「会社A」「会社X」「会社B」に関する情報を「その他の情報」として持っている。

【0032】DSAプロセス11は、第1の候補である

下位エントリ21の属性値“会社A”と、目的エントリの識別名(DN)の2番目の相対識別名(RDN)の属性値“会社X”とを比較する(ステップA2)。その結果は不一致であるため、次の下位エントリ22を取り出す(ステップA4)。

【0033】次に、下位エントリ22の属性値“会社X”と、目的エントリの識別名の2番目の相対識別名(RDN)の属性値“会社X”を比較する(ステップA2)。その結果は一致であるため、下位エントリ22の属性型“組織名”と目的エントリの識別名の2番目の相対識別名(RDN)の属性型“組織名”を比較する(ステップA3)。

【0034】その結果も一致であるため、目的エントリは下位エントリ22の下位にあることが判別される。これで識別名{“国名”、“日本”}、{“組織名”、“会社X”}、{“一般名”、“橋本浩一”}のうち{“国名”、“日本”}、{“組織名”、“会社X”}までたどることができる。

【0035】従来のように属性型から比較すると、前述した状況では2回の属性型の比較と2回の属性値の比較で、計4回の比較が必要であるが、図3のフローチャートに示す手順によれば、1回の属性型の比較と2回の属性値の比較で計3回の比較だけで必要な結果が得られる。

【0036】以下、同様にして、下位エントリの候補について、図3に示すフローチャートに従って、相対識別名(RDN)の比較処理と、識別名(DN)にある相対識別名(RDN)の個数分だけ繰り返すことによって、{“国名”、“日本”}、{“組織名”、“会社X”}、{“一般名”、“橋本浩一”}までたどり、目的エントリを取り出すことができる。

【0037】以上のような処理を行なうことにより、全体の比較回数を減らすことによって、検索速度を向上させることになる。次に、ハッシュ演算器12を用いる場合の動作について説明する。

【0038】まず、利用者が検索したい目的エントリの識別名(DN)が、DUAプロセス20、通信回線19を介してDSAプロセス11に送信される。DSAプロセス11は、利用者が検索したいエントリの識別名(DN)を受信すると、相対識別名(RDN)の属性値をハッシュ演算器12に出力し、そこで求められたハッシュ値と共に識別名をメモリ13に書き込む。

【0039】図5は、DSAプロセス11によってメモリ13に書き込まれている、目的エントリに対応する識別名(DN)の一例を示している。識別名(DN)は、図2に示す相対識別名(RDN)の並びに、それぞれハッシュ値が付加される。

【0040】次に、図6に示すフローチャートを参照しながら、DIB17aから目的エントリを取り出す処理について説明する。この際、図3にフローチャートを参

照しながら説明した比較処理に、さらに属性値のハッシュ値の比較処理が付加されたものである。

【0041】DSAプロセス11は、ドライバ16を使って、ファイル17のDIB17aから「日本」を示すエントリの、全ての下位エントリの候補をファイル17からメモリ13に読み出す(ステップB1)。

【0042】図7は、メモリ13に格納された下位エントリ31、32、33の候補を示し、「日本」を示すエントリの下位エントリがドライバ16によってファイル17から読み出されている状態を示している。

【0043】図7に示すように、ファイル17にDIB17aのエントリを格納する際には、相対識別名(RDN)の属性値のハッシュ値も格納する。すなわち、下位エントリ31、32、33は、相対識別名(RDN)の属性値のハッシュ値が付加されており、それぞれ{“組織名”、“会社A”、“53”}、{“組織名”、“会社X”、“77”}、{“組織名”、“会社B”、“49”}である。

【0044】DSAプロセス11は、下位エントリの属性値と目的エントリの識別名の相対識別名(RDN)の属性値を比較する前に、各々の属性値のハッシュ値の比較を行なう(ステップB2)。

【0045】DSAプロセス11は、第1の候補である下位エントリ31のハッシュ値“53”と、目的エントリの識別名(DN)の2番目の相対識別名(RDN)のハッシュ値“77”とを比較する(ステップB2)。その結果は不一致であるため、次の下位エントリ32を取り出す(ステップB5)。

【0046】次に、下位エントリ32のハッシュ値“77”と、目的エントリの識別名の2番目の相対識別名(RDN)のハッシュ値“77”を比較する(ステップB2)。その結果は一致であるため、以下、図3のフローチャートに示したステップA2、A3と同様にして、属性値の比較(ステップB3)、属性型の比較(ステップB4)を行ない、その比較結果に基づいて必要な結果を得る。

【0047】図6に示すフローチャートの手順によれば、属性値の比較は2段階になるが、通常の属性値の長さは数十バイトになるため、ハッシュ値の長さを例えば1バイト程度にすれば、少ない比較処理時間によって候補の絞り込みができる。従って、結果として、属性値の比較にかかる処理時間を短縮することができ、検索速度を向上させることになる。

【0048】次に、図6に示すフローチャートに従う処理において検索速度を向上させるために、ドライバ16に設けられる機能について説明する。ドライバ16は、下位エントリ31、32、33をファイル17からメモリ13に読み出す際に、図8(a)に示すような、下位エントリ一覧テーブル41を作成してメモリ13に格納する。

【0049】下位エントリ一覧テーブル31は、各下位エントリ31、32、33の相対識別名(RDN)の属性値のハッシュ値と、それぞれに対応する下位エントリ31、32、33を示すアドレスをもっている。

【0050】ドライバ16は、図6のフローチャートに従って相対識別名(RDN)の比較をするときに、下位エントリ一覧テーブル41を参照することで、図8

(b)に示す下位エントリを高速に取り出すことができる。

【0051】このようにして、ディレクトリシステムを実現する情報処理装置がエントリを検索するために相対識別名(RDN)の比較を行なうときに、属性型の比較の前に属性値の比較を行なうことにより、属性値の比較で一致したエントリについてのみ属性型を比較することで、全体の比較回数を減らすことができ、検索速度を向上させることができる。

【0052】また、ハッシュ演算器12によって目的エントリの識別名(DN)の属性値に固定長のハッシュ値をもたせ、このハッシュ値によって検索を行なうことから、一致するものの中で属性値の比較を行なうことで、通常、数十バイトの長さとなるために処理時間を多く要する属性値の比較を少なくすることができるので、2段階の属性値の比較となったとしても、全体としての検索速度を向上させることができる。

【0053】また、下位エントリの候補である会社A、会社Xと会社Bの相対識別名(RDN)を高速に取り出すために下位エントリ一覧テーブル41を使用することによって、高速に下位エントリにアクセスすることができる。全体としての検索速度を向上させることができる。

【0054】

【発明の効果】以上のように本発明によれば、エントリを検索するために相対識別名(RDN)の比較を行なうときに、属性値の比較で一致したものだけに属性型を比較すること、また属性値に固定長のハッシュ値をもたせ、その値による検索を行ってから、一致するものの中で属性値の比較を行なうこと、さらに下位エントリの候

補である相対識別名(RDN)を高速に取り出すための下位エントリ一覧テーブルを使うことで、ディレクトリ情報ベースに対する検索速度を向上させることが可能となるものである。

【図面の簡単な説明】

【図1】本発明の一実施例に係わる情報処理装置の構成を示すブロック図。

【図2】本実施例におけるメモリ13に格納されている検索の対象となる目的エントリの識別名の一例を示す図。

【図3】本実施例におけるDIB17aから目的エントリを取り出す処理について説明するためのフローチャート。

【図4】本実施例におけるメモリ13に格納されている下位エントリの候補の一例を示す図。

【図5】本実施例におけるメモリ13に格納されている検索の対象となる目的エントリの識別名にハッシュ値が付加された一例を示す図。

【図6】本実施例における属性値のハッシュ値の比較が付加されたDIB17aから目的エントリを取り出す処理について説明するためのフローチャート。

【図7】本実施例におけるメモリ13に格納されている属性値のハッシュ値が付加された下位エントリの候補の一例を示す図。

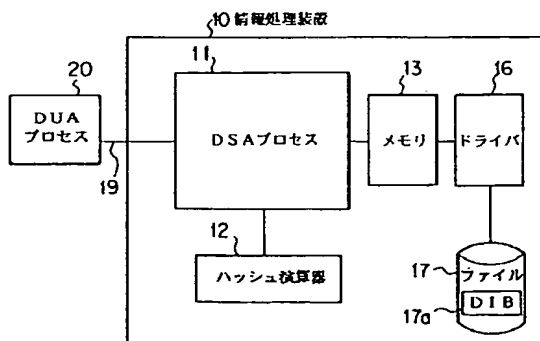
【図8】本実施例におけるメモリ13に格納されている下位エントリ一覧テーブル41の一例を示す図。

【図9】ディレクトリ情報ベース(DIB)のエントリの一例を示す図。

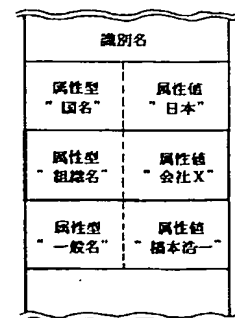
【符号の説明】

10…情報処理装置、11…DSA(ディレクトリサービスエージェント)プロセス、12…ハッシュ演算器、13…メモリ、16…ドライバ、17…ファイル、17a…ディレクトリ情報ベース(DIB)、19…通信回線、20…DUAプロセス、21、22、23、31、32、33…下位エントリの情報、41…下位エントリ一覧テーブル。

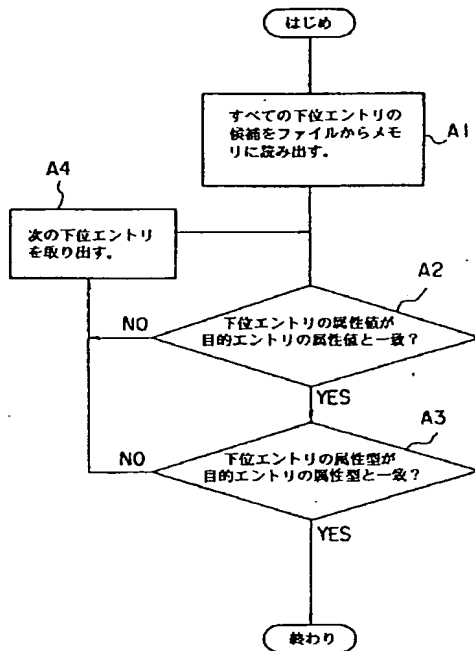
【図1】



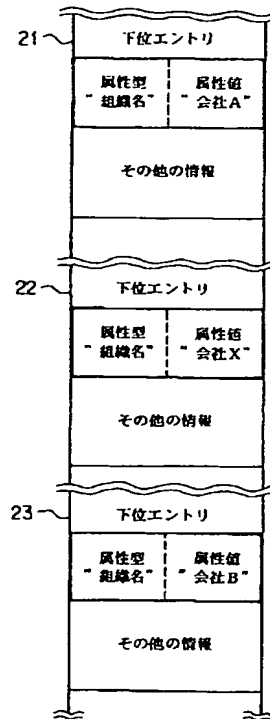
【図2】



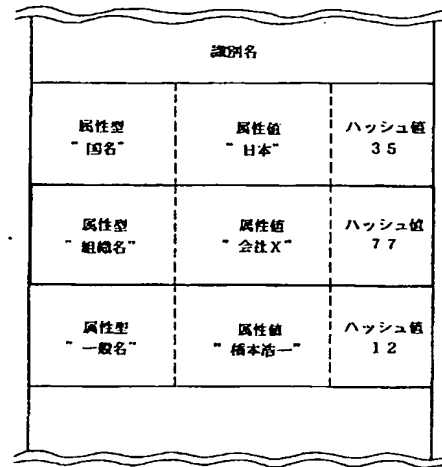
【図3】



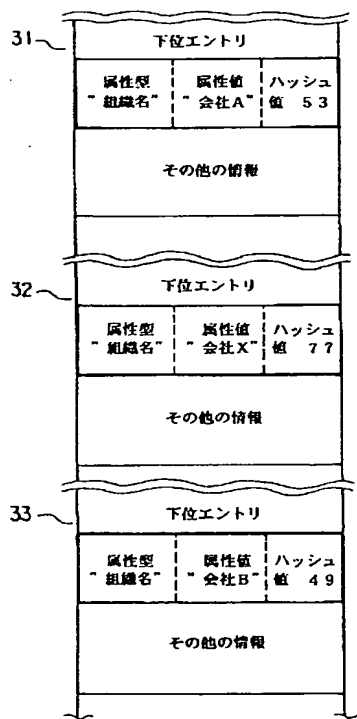
【図4】



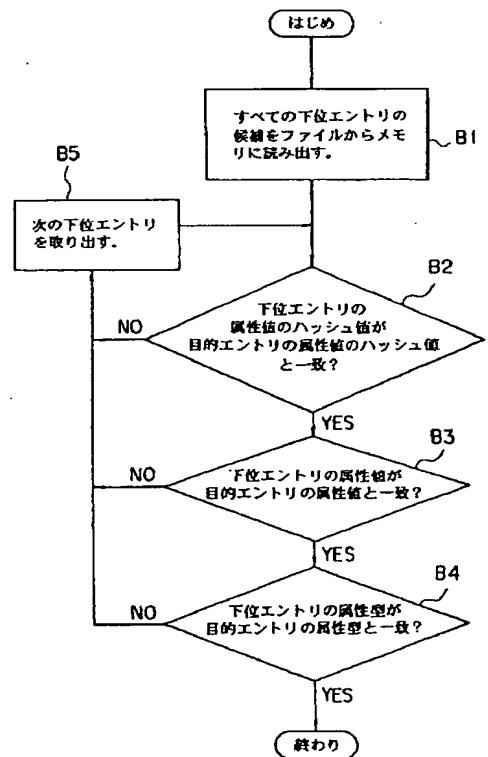
【図5】



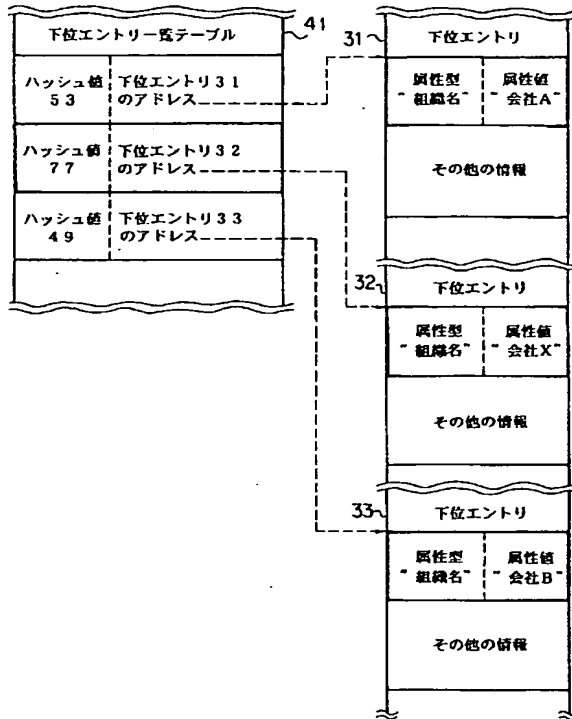
【図7】



【図6】



【図8】



【図9】

